Assignment 1

Total: 35pts

Part A - Write Code! (5pts)

For each item below, determine the appropriate TypeScript code to generate the desired output.

---

1. Display your name and student ID to the console.

In [1]:
```
console.log("Tonglin Yang W0526437")
```

```
Tonglin Yang W0526437
```

Example Output:
Jane Doe W0123456

---

2. Create a variable, called **num**, to hold the integer value, **10**. Then, display the contents of the variable.

In [2]:
```
let num: number=10;
console.log(num);
```

```
10
```

Expected Output:
10

---

3. Create a variable to hold the name of your favourite TV character. Then, use a template literal to display the message "My favourite TV star is: " with the variable contents appended. (e.g. "My favourite TV star is: Homer Simpson")

In [58]:
```
let person = "Shizuki";
console.log(`My favourite TV star is: ${person}`);
```

```
My favourite TV star is: Shizuki
```

Example Output: My favourite TV Star is : Homer Simpson

4. Display Santa Clause's catch phrase (Ho Ho Ho) using the string repeat method.

In [5]:
```
console.log("Ho".repeat(3).trim())
```

```
HoHoHo
```

Expected Output:
Ho Ho Ho

---

5. Create the TypeScript statement to calculate the following mathematical expression: $$ \begin{align*} \frac {2 + 3} 2 \times \frac {6 - 4} 3 \end{align*} $$ Round the results to three decimal points.

In [6]:
```
let result: number=(2+3)*(6-4)/(2*3);
console.log(result.toFixed(3));
```

```
1.667
```

Expected Output:
1.667

Part B - Fix the Error! (15pts)

Listed below are several snippets of code that contain small errors. For each of the questions, copy the given code and **correct the error** so the code will generate the desired output.

NOTE: Don't add any additional lines or entirely rewrite the given code, just correct the error to make the line(s) work as expected. You will be marked incorrect otherwise.

---

6. Display a string of text on the screen.

```
// DON'T modify this code
console.log "This is a story about a man named Jed."
```

In [7]:
```
// DO enter your answer below this line
console.log ("This is a story about a man named Jed.")
```

```
This is a story about a man named Jed.
```

Expected Output:
This is a story about a man named Jed.

---

7. Display a string and a decimal number using string concatentation.

```
// DON'T modify this code
console.log("PI = " 3.14159)
```

In [8]:
```
// DO enter your answer below this line
console.log("PI =  3.14159")
```

```
PI =  3.14159
```

Expected Output:

PI = 3.14159

---

8. Fix the line so that the answer is correctly calculated and displayed.

```
// DON'T modify this code
console.log("1 + 2")
```

In [9]:
```
// DO enter your answer below this line
console.log(1 + 2)
```

```
3
```

Expected Output:

3

---

9. Correct the following calculation so that the output makes sense.

```
// DON'T modify this code
console.log("3 cubed =", 3 * 3)
```

In [10]:
```
// DO enter your answer below this line
console.log("3 cubed =", 3 ** 3)
```

```
3 cubed = 27
```

Expected Output:

3 cubed = 27

---

10. Fix the template literal to display a name in the format *<last name>, <first name>*

```
// DON'T modify this code
let firstName = "John"
let lastName = "Smith"
console.log('${lastName}, ${firstName}')
```

In [14]:
```
// DO enter your answer below this line
let firstName = "John"
```

```
let lastName = "Smith"
console.log(`${lastName}, ${firstName}`)
```

```
Smith, John
```

Expected Output:
Smith, John

---

11. Display the answer of an integer division and the remainder. There should not be any decimal points displayed.

```
// DON'T modify this code
console.log("193 / 5 =", 193/5, "remainder", 193/5)
```

In [13]:
```
// DO enter your answer below this line
console.log("193 / 5 =", Math.floor(193/5), "remainder", 193%5)
```

```
193 / 5 = 38 remainder 3
```

Expected Output:
193 / 5 = 38 remainder 3

---

12. Correct the following calculation to convert 100° Celsius to Fahrenheit.

```
// DON'T modify this code
let temperature = 100.0
console.log(temperature + 32 * 9 / 5)
```

In [15]:
```
// DO enter your answer below this line
let temperature = 100.0
console.log((temperature* 9 / 5)+32)
```

```
212
```

Expected Output:
212

---

13. Correct the following to calculate and display the expected answer using string concatentation.

```
// DON'T modify this code
console.log("1 + 2 = " + 1 + 2)
```

In [16]:
```
// DO enter your answer below this line
console.log("1 + 2 = " + (1 + 2))
```

```
1 + 2 = 3
```

Expected Output:
1 + 2 = 3

---

14. Correct the error using best practices to display the desired output.

```
// DON'T modify this code
let ninety-nine = 99
console.log("Number = " + ninety-nine)
```

In [19]:
```
// DO enter your answer below this line
let ninetynine = 99
console.log("Number = " + ninetynine)
```

```
Number = 99
```

Expected Output:
Number = 99

---

15. Use a single print statement to print on multiple lines.

```
// DON'T modify this code
console.log("This is on multiple lines.")
```

In [21]:
```
// DO enter your answer below this line
console.log("This is\non multiple\nlines.")
```

```
This is
on multiple
lines.
```

Expected Output:
This is
on multiple
lines.

---

16. Comment out the first line.

```
// DON'T modify this code
console.log("This is just a comment.")
console.log("You should only see this line.")
```

In [22]:
```
// DO enter your answer below this line
//console.log("This is just a comment.")
console.log("You should only see this line.")
```

```
You should only see this line.
```

Expected Output:
You should only see this line.

---

17. Make some ASCII art! Correct the string to create the expected output.

```
// DON'T modify this code
console.log(".     . \ O /  - -   |   ^  / \ -   -")
```

In [42]:
```
// DO enter your answer below this line
console.log(" .     .\n  \\ O /   \n   - -\n    |\n    ^ \n   / \\  \n  -   -")
```

```
.     .
 \ O /
  - -
   |
   ^
  / \
 -   -
```

Expected Output :

```
.     .
 \ O /
  - -
   |
   ^
  / \
 -   -
```

---

18. A couple of problems here.... (Remember, just fix the errors, don't rewrite the statement. There are two errors!)

```
// DON'T modify this code
let not = "not"
console.log(`Something's $(not) right ` - "here.")
```

In [45]:
```
// DO enter your answer below this line
let not = "not"
console.log(`Something's ${not} right  here.`)
```

```
Something's not right  here.
```

Expected Output:
Something's not right here.

---

19. Fix the following bill calculation by adding a 15% gratuity.

```
// DON'T modify this code
let restaurantBill = 10.00
// add 15% to the bill here
console.log('Total: $' + restaurantBill.toFixed(2))
```

In [54]:
```
// DO enter your answer below this line
let restaurantBill = 10.00
// add 15% to the bill here
console.log('Total: $' + (restaurantBill*1.15).toFixed(2))
```

```
Total: $11.50
```

Expected Output:
Total: $11.50

---

20. Correct the following inventory count by subtracting the number of items sold from the total.

```
// DON'T modify this code
let totalItems = 100
let itemsSold = 42
// correct the total here
console.log(`Sold: ${itemsSold}, Remaining: ${totalItems}`)
```

In [53]:
```
// DO enter your answer below this line
let totalItems = 100
let itemsSold = 42
// correct the total here
console.log(`Sold: ${itemsSold}, Remaining: ${totalItems-itemsSold}`)
```

```
Sold: 42, Remaining: 58
```

Expected Output:
Sold: 42, Remaining: 58

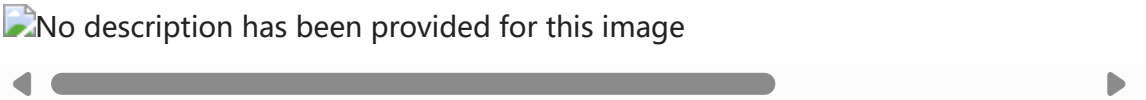
Part C - Script (15pts)

Create a TypeScript program that calculates the roots of a quadratic equation using the equation, $x = \large {-b \pm \sqrt{b^2 - 4ac} \over 2a}$.

You can assume that the equations used will always have two defined roots. (i.e. ${a \ne 0}$)

The program should allow the user to input the three variables of the formula, a, b, and c, and then output the resulting roots.

Example:



In [55]:
```
// Part C - Quadratic Equation Roots Calculator

let a: number = 1;
let b: number = -3;
let c: number = 2;

let discriminant: number = b * b - 4 * a * c;

let root1: number = (-b + Math.sqrt(discriminant)) / (2 * a);
let root2: number = (-b - Math.sqrt(discriminant)) / (2 * a);

console.log(`The roots are: ${root1} and ${root2}`);
```

```
The roots are: 2 and 1
```

In [ ]:
```

```